Technical Report for Data storage assignment

Name:

Mohankumar T

Aarti Sharma

Ajay kumar

Student No

10547112

10548601

10546004

Contents

[1. INTRODUCTION 1](#_Toc37725667)

[1.2. REASONS FOR SELECTING THE SUBJECT AREA AND DATA 1](#_Toc37725668)

[1.3. VISION AND GOALS 1](#_Toc37725669)

[1.4. KEY STAKEHOLDERS 1](#_Toc37725670)

[1.5. BUSINESS REQUIREMENTS 1](#_Toc37725671)

[2. SCHEMA 1](#_Toc37725672)

[2.1 Data warehouse: data model: Star Schema 1](#_Toc37725673)

[2.1.1 Physical Data Model: 1](#_Toc37725674)

[2.1.2 Tables list: 1](#_Toc37725675)

[2.1.3 Columns Information: 1](#_Toc37725676)

[2.2 Implementation in MS SQL (Database) 1](#_Toc37725677)

[3. ETL - Implementation in SSIS (ETL integration) 1](#_Toc37725678)

[3. 1 Overall Flow: 1](#_Toc37725679)

[3.2 Individual load Flow: 1](#_Toc37725680)

[4. Reports and Visualization 1](#_Toc37725681)

[4.1 Implementation in SSRS 1](#_Toc37725682)

[4. 1.1 Transactions made in bank 1](#_Toc37725683)

[4.1.2 Loans per tenure in months: 1](#_Toc37725684)

[4.1.3 Credit card type issued to customer 1](#_Toc37725685)

[4.1.4 Worst affected in low income group: 1](#_Toc37725686)

[4.2 Implementation in Tableau 1](#_Toc37725687)

[4.2.1 Tableau implementation 1: Loan types availed over calender 1](#_Toc37725688)

[4.2.2 Tableau implementation 2: Geography of branches 1](#_Toc37725689)

[4.2.3 Tableau implementation 3: loan amount granted based on purpose 1](#_Toc37725690)

[4.2.4 Tableau implementation 4: loan payments made 1](#_Toc37725691)

[5. Graph database: neo4j & cypher queries 1](#_Toc37725692)

[5.1 –relation ship between job title and gender 1](#_Toc37725693)

[5.2 –card type subscriptions 1](#_Toc37725694)

[5.3 –delivery location to country relationship 1](#_Toc37725695)

[5.4 –Discount Type to category comparison 1](#_Toc37725696)

[6. Conclusion 1](#_Toc37725697)

[7. Bibliography 1](#_Toc37725698)

[APPENDIX A – neo4j CODE 1](#_Toc37725699)

# 1. INTRODUCTION

Banking Dataset contains real world data source from Czech banks data. All the personally identifiable information is masked to avoid identity exposure. Addresses are not real but masked random readable information. The dataset contains information on the loan products availed by various demographics and the usage of loan accounts. Along with the account holder’s information and the type of credit card offered to the account individual based on various parameters. We have analyzed the dataset after building a data model, using SSRS, R and Neo4J queries.

## 1.2. REASONS FOR SELECTING THE SUBJECT AREA AND DATA

The Core banking subject area provides various details related to the customer transactions, loan book and account information. The data set can be used to understand backgrounds of various account holders coming from various demographics and having various opportunities in life. It is useful to understand the lifestyle activities through the transactions data and the credit limit provided to various individuals based on their credit profile.

## 1.3. VISION AND GOALS

To visualize the core banking data set after applying various data modelling techniques and loading the data to the data warehouse through SSIS integration tool. To visualize the data to gain insights using SSRS, R and Neo4J tools.

## 1.4. KEY STAKEHOLDERS

## 1.5. BUSINESS REQUIREMENTS

🡪 recent version of MSBI for MSSQL

🡪 SSRS and SSIS

🡪 Neo4J and R

# 2. SCHEMA

# 2.1 Data warehouse: data model: Star Schema

## 2.1.1 Physical Data Model:

The Below data model is constructed from various input files from the source.

A screenshot of a computer

Description automatically generated

## 2.1.2 Tables list:

|  |
| --- |
| **Name** |
| Account\_Fact |
| Card\_Fact |
| CardType\_Dim |
| AcctFreq\_Dim |
| client\_Fact |
| Area\_Dim |
| Gender\_Dim |
| Disposition\_fact |
| District\_dim |
| ZipCode\_Districts\_fact |
| Loan\_fact |
| LoanPurpose\_Dim |
| loan\_to\_Account |
| Transaction\_order\_fact |
| account\_to\_order |

## 2.1.3 Columns Information:

|  |  |  |
| --- | --- | --- |
| **Table** | **Attribute** | **Data Type** |
| Account\_Fact | account\_id | Varchar(max) |
| Account\_Fact | district\_id | Int |
| Account\_Fact | parseddate | Date |
| Account\_Fact | year | Int |
| Account\_Fact | month | Int |
| Account\_Fact | day | Int |
| Account\_Fact | freq\_id | Int |
| Card\_Fact | card\_id | Varchar(max) |
| Card\_Fact | disp\_id | Varchar(max) |
| Card\_Fact | year | Int |
| Card\_Fact | month | Int |
| Card\_Fact | day | Int |
| Card\_Fact | fulldate | Date |
| Card\_Fact | Type\_ID | Int |
| CardType\_Dim | Type\_ID | Int |
| CardType\_Dim | Type | Varchar(max) |
| AcctFreq\_Dim | freq\_id | Int |
| AcctFreq\_Dim | frequency | Varchar(max) |
| client\_Fact | client\_id | Char(%p1) |
| client\_Fact | fulldate | Char(%p1) |
| client\_Fact | day | Char(%p1) |
| client\_Fact | month | Char(%p1) |
| client\_Fact | year | Char(%p1) |
| client\_Fact | age | Char(%p1) |
| client\_Fact | social | Char(%p1) |
| client\_Fact | first | Char(%p1) |
| client\_Fact | middle | Char(%p1) |
| client\_Fact | last | Char(%p1) |
| client\_Fact | phone | Char(%p1) |
| client\_Fact | email | Char(%p1) |
| client\_Fact | address\_1 | Char(%p1) |
| client\_Fact | address\_2 | Char(%p1) |
| client\_Fact | district\_id | Char(%p1) |
| client\_Fact | zipcode | Varchar(max) |
| client\_Fact | Gender\_ID | Int |
| Area\_Dim | zipcode | Varchar(max) |
| Area\_Dim | state | Varchar(max) |
| Area\_Dim | city | Varchar(max) |
| Area\_Dim | district\_id | Int |
| Gender\_Dim | Gender\_ID | Int |
| Gender\_Dim | Gender | Varchar(max) |
| Disposition\_fact | disp\_id | Varchar(max) |
| Disposition\_fact | type | Varchar(max) |
| Disposition\_fact | client\_id | Varchar(max) |
| Disposition\_fact | account\_id | Varchar(max) |
| District\_dim | district\_id | Int |
| District\_dim | city | Varchar(max) |
| District\_dim | state\_name | Varchar(max) |
| District\_dim | state\_abbrev | Varchar(max) |
| District\_dim | region | Varchar(max) |
| District\_dim | division | Varchar(max) |
| ZipCode\_Districts\_fact | district\_id | Int |
| ZipCode\_Districts\_fact | zipcode | Varchar(max) |
| Loan\_fact | loan\_id | Varchar(max) |
| Loan\_fact | purpose\_ID | Int |
| Loan\_fact | account\_id | Varchar(max) |
| Loan\_fact | amount | Float |
| Loan\_fact | duration | Int |
| Loan\_fact | payments | Float |
| Loan\_fact | status | Varchar(max) |
| Loan\_fact | year | Int |
| Loan\_fact | month | Int |
| Loan\_fact | day | Int |
| Loan\_fact | fulldate | Date |
| Loan\_fact | location | Int |
| Loan\_fact | purpose | Varchar(max) |
| LoanPurpose\_Dim | purpose\_ID | Int |
| LoanPurpose\_Dim | purpose | Varchar(max) |
| loan\_to\_Account | account\_id | Varchar(max) |
| loan\_to\_Account | freq\_id | Int |
| loan\_to\_Account | loan\_id | Varchar(max) |
| loan\_to\_Account | purpose\_ID | Int |
| Transaction\_order\_fact | order\_id | Varchar(max) |
| Transaction\_order\_fact | account\_id | Varchar(max) |
| Transaction\_order\_fact | bank\_to | Varchar(max) |
| Transaction\_order\_fact | account\_to | Varchar(max) |
| Transaction\_order\_fact | amount | Float |
| Transaction\_order\_fact | k\_symbol | Varchar(max) |
| account\_to\_order | account\_id | Varchar(max) |
| account\_to\_order | freq\_id | Int |
| account\_to\_order | order\_id | Varchar(max) |

# 2.2 Implementation in MS SQL (Database)

-- Create tables section -------------------------------------------------

-- Table Account\_Fact

CREATE TABLE [Account\_Fact]

(

[account\_id] Varchar(max) NOT NULL,

[district\_id] Int NULL,

[parseddate] Date NULL,

[year] Int NULL,

[month] Int NULL,

[day] Int NULL,

[freq\_id] Int NOT NULL

)

go

-- Add keys for table Account\_Fact

ALTER TABLE [Account\_Fact] ADD CONSTRAINT [PK\_Account\_Fact] PRIMARY KEY ([account\_id],[freq\_id])

go

-- Table Card\_Fact

CREATE TABLE [Card\_Fact]

(

[card\_id] Varchar(max) NOT NULL,

[disp\_id] Varchar(max) NULL,

[year] Int NULL,

[month] Int NULL,

[day] Int NULL,

[fulldate] Date NULL,

[Type\_ID] Int NOT NULL

)

go

-- Add keys for table Card\_Fact

ALTER TABLE [Card\_Fact] ADD CONSTRAINT [PK\_Card\_Fact] PRIMARY KEY ([card\_id],[Type\_ID])

go

-- Table CardType\_Dim

CREATE TABLE [CardType\_Dim]

(

[Type\_ID] Int NOT NULL,

[Type] Varchar(max) NULL

)

go

-- Add keys for table CardType\_Dim

ALTER TABLE [CardType\_Dim] ADD CONSTRAINT [PK\_CardType\_Dim] PRIMARY KEY ([Type\_ID])

go

-- Table AcctFreq\_Dim

CREATE TABLE [AcctFreq\_Dim]

(

[freq\_id] Int NOT NULL,

[frequency] Varchar(max) NULL

)

go

-- Add keys for table AcctFreq\_Dim

ALTER TABLE [AcctFreq\_Dim] ADD CONSTRAINT [PK\_AcctFreq\_Dim] PRIMARY KEY ([freq\_id])

go

-- Table client\_Fact

CREATE TABLE [client\_Fact]

(

[client\_id] Char(1) NOT NULL,

[fulldate] Char(1) NULL,

[day] Char(1) NULL,

[month] Char(1) NULL,

[year] Char(1) NULL,

[age] Char(1) NULL,

[social] Char(1) NULL,

[first] Char(1) NULL,

[middle] Char(1) NULL,

[last] Char(1) NULL,

[phone] Char(1) NULL,

[email] Char(1) NULL,

[address\_1] Char(1) NULL,

[address\_2] Char(1) NULL,

[district\_id] Char(1) NULL,

[zipcode] Varchar(max) NOT NULL,

[Gender\_ID] Int NOT NULL

)

go

-- Add keys for table client\_Fact

ALTER TABLE [client\_Fact] ADD CONSTRAINT [PK\_client\_Fact] PRIMARY KEY ([client\_id],[zipcode],[Gender\_ID])

go

-- Table Area\_Dim

CREATE TABLE [Area\_Dim]

(

[zipcode] Varchar(max) NOT NULL,

[state] Varchar(max) NULL,

[city] Varchar(max) NULL,

[district\_id] Int NOT NULL

)

go

-- Add keys for table Area\_Dim

ALTER TABLE [Area\_Dim] ADD CONSTRAINT [PK\_Area\_Dim] PRIMARY KEY ([zipcode])

go

-- Table Gender\_Dim

CREATE TABLE [Gender\_Dim]

(

[Gender\_ID] Int NOT NULL,

[Gender] Varchar(max) NULL

)

go

-- Add keys for table Gender\_Dim

ALTER TABLE [Gender\_Dim] ADD CONSTRAINT [PK\_Gender\_Dim] PRIMARY KEY ([Gender\_ID])

go

-- Table Disposition\_fact

CREATE TABLE [Disposition\_fact]

(

[disp\_id] Varchar(max) NOT NULL,

[type] Varchar(max) NULL,

[client\_id] Varchar(max) NULL,

[account\_id] Varchar(max) NULL

)

go

-- Add keys for table Disposition\_fact

ALTER TABLE [Disposition\_fact] ADD CONSTRAINT [PK\_Disposition\_fact] PRIMARY KEY ([disp\_id])

go

-- Table District\_dim

CREATE TABLE [District\_dim]

(

[district\_id] Int NOT NULL,

[city] Varchar(max) NULL,

[state\_name] Varchar(max) NULL,

[state\_abbrev] Varchar(max) NULL,

[region] Varchar(max) NULL,

[division] Varchar(max) NULL

)

go

-- Add keys for table District\_dim

ALTER TABLE [District\_dim] ADD CONSTRAINT [PK\_District\_dim] PRIMARY KEY ([district\_id])

go

-- Table ZipCode\_Districts\_fact

CREATE TABLE [ZipCode\_Districts\_fact]

(

[district\_id] Int NOT NULL,

[zipcode] Varchar(max) NOT NULL

)

go

-- Add keys for table ZipCode\_Districts\_fact

ALTER TABLE [ZipCode\_Districts\_fact] ADD CONSTRAINT [PK\_ZipCode\_Districts\_fact] PRIMARY KEY ([district\_id],[zipcode])

go

-- Table Loan\_fact

CREATE TABLE [Loan\_fact]

(

[loan\_id] Varchar(max) NOT NULL,

[purpose\_ID] Int NOT NULL,

[account\_id] Varchar(max) NULL,

[amount] Float NULL,

[duration] Int NULL,

[payments] Float NULL,

[status] Varchar(max) NULL,

[year] Int NULL,

[month] Int NULL,

[day] Int NULL,

[fulldate] Date NULL,

[location] Int NULL,

[purpose] Varchar(max) NULL

)

go

-- Add keys for table Loan\_fact

ALTER TABLE [Loan\_fact] ADD CONSTRAINT [PK\_Loan\_fact] PRIMARY KEY ([loan\_id],[purpose\_ID])

go

-- Table LoanPurpose\_Dim

CREATE TABLE [LoanPurpose\_Dim]

(

[purpose\_ID] Int NOT NULL,

[purpose] Varchar(max) NULL

)

go

-- Add keys for table LoanPurpose\_Dim

ALTER TABLE [LoanPurpose\_Dim] ADD CONSTRAINT [PK\_LoanPurpose\_Dim] PRIMARY KEY ([purpose\_ID])

go

-- Table loan\_to\_Account

CREATE TABLE [loan\_to\_Account]

(

[account\_id] Varchar(max) NOT NULL,

[freq\_id] Int NOT NULL,

[loan\_id] Varchar(max) NOT NULL,

[purpose\_ID] Int NOT NULL

)

go

-- Add keys for table loan\_to\_Account

ALTER TABLE [loan\_to\_Account] ADD CONSTRAINT [PK\_loan\_to\_Account] PRIMARY KEY ([account\_id],[freq\_id],[loan\_id],[purpose\_ID])

go

-- Table Transaction\_order\_fact

CREATE TABLE [Transaction\_order\_fact]

(

[order\_id] Varchar(max) NOT NULL,

[account\_id] Varchar(max) NULL,

[bank\_to] Varchar(max) NULL,

[account\_to] Varchar(max) NULL,

[amount] Float NULL,

[k\_symbol] Varchar(max) NULL

)

go

-- Add keys for table Transaction\_order\_fact

ALTER TABLE [Transaction\_order\_fact] ADD CONSTRAINT [PK\_Transaction\_order\_fact] PRIMARY KEY ([order\_id])

go

-- Table account\_to\_order

CREATE TABLE [account\_to\_order]

(

[account\_id] Varchar(max) NOT NULL,

[freq\_id] Int NOT NULL,

[order\_id] Varchar(max) NOT NULL

)

go

-- Add keys for table account\_to\_order

ALTER TABLE [account\_to\_order] ADD CONSTRAINT [PK\_account\_to\_order] PRIMARY KEY ([account\_id],[freq\_id],[order\_id])

go

-- Create foreign keys (relationships) section -------------------------------------------------

ALTER TABLE [Card\_Fact] ADD CONSTRAINT [Relationship1] FOREIGN KEY ([Type\_ID]) REFERENCES [CardType\_Dim] ([Type\_ID]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Account\_Fact] ADD CONSTRAINT [Relationship2] FOREIGN KEY ([freq\_id]) REFERENCES [AcctFreq\_Dim] ([freq\_id]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [client\_Fact] ADD CONSTRAINT [Relationship3] FOREIGN KEY ([zipcode]) REFERENCES [Area\_Dim] ([zipcode]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [client\_Fact] ADD CONSTRAINT [Relationship4] FOREIGN KEY ([Gender\_ID]) REFERENCES [Gender\_Dim] ([Gender\_ID]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [ZipCode\_Districts\_fact] ADD CONSTRAINT [Relationship14] FOREIGN KEY ([district\_id]) REFERENCES [District\_dim] ([district\_id]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [ZipCode\_Districts\_fact] ADD CONSTRAINT [Relationship15] FOREIGN KEY ([zipcode]) REFERENCES [Area\_Dim] ([zipcode]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Loan\_fact] ADD CONSTRAINT [Relationship16] FOREIGN KEY ([purpose\_ID]) REFERENCES [LoanPurpose\_Dim] ([purpose\_ID]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [loan\_to\_Account] ADD CONSTRAINT [Relationship17] FOREIGN KEY ([account\_id], [freq\_id]) REFERENCES [Account\_Fact] ([account\_id], [freq\_id]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [loan\_to\_Account] ADD CONSTRAINT [Relationship18] FOREIGN KEY ([loan\_id], [purpose\_ID]) REFERENCES [Loan\_fact] ([loan\_id], [purpose\_ID]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [account\_to\_order] ADD CONSTRAINT [Relationship19] FOREIGN KEY ([account\_id], [freq\_id]) REFERENCES [Account\_Fact] ([account\_id], [freq\_id]) ON UPDATE NO ACTION ON DELETE NO ACTION

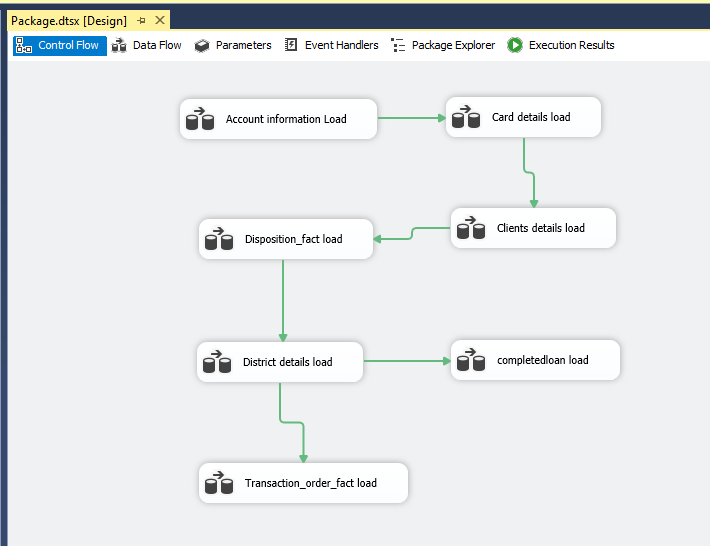
go

ALTER TABLE [account\_to\_order] ADD CONSTRAINT [Relationship20] FOREIGN KEY ([order\_id]) REFERENCES [Transaction\_order\_fact] ([order\_id]) ON UPDATE NO ACTION ON DELETE NO ACTION

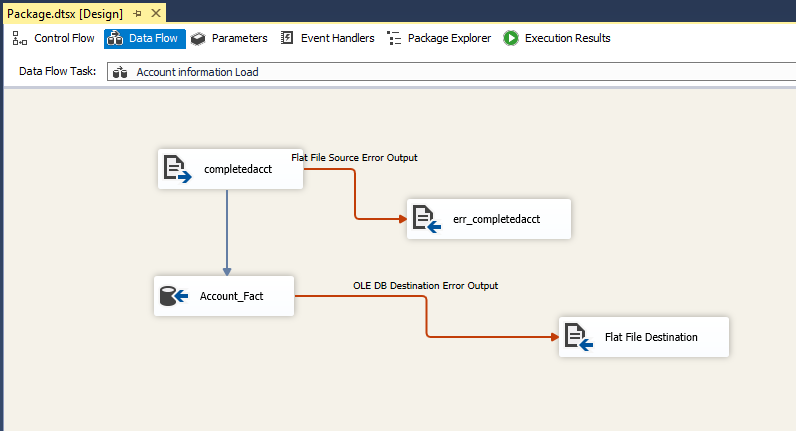
go

# 3. ETL - Implementation in SSIS (ETL integration)

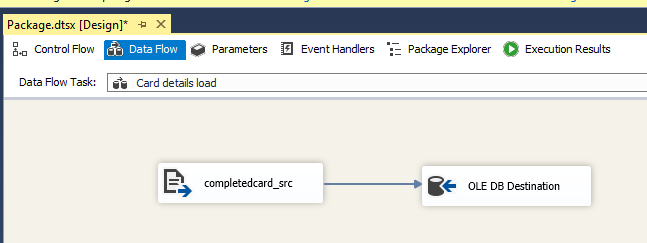
3. 1 Overall Flow:

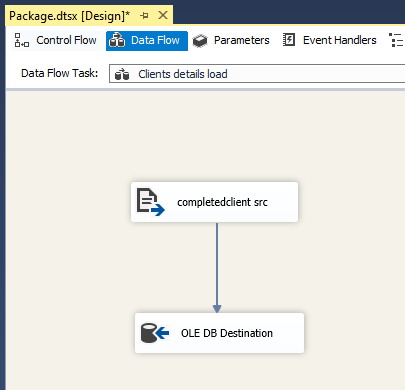


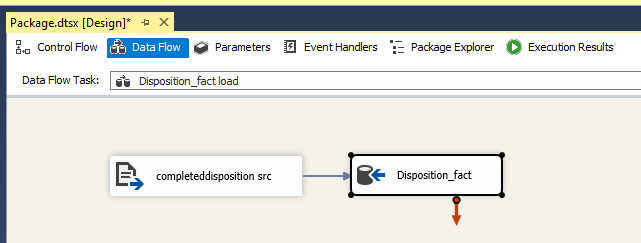
3.2 Individual load Flow:

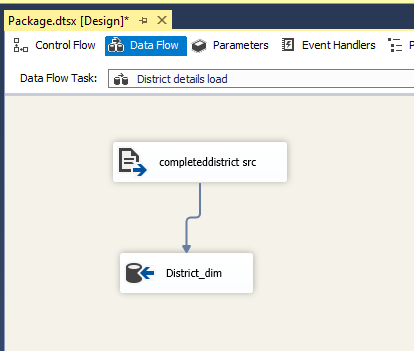


# 





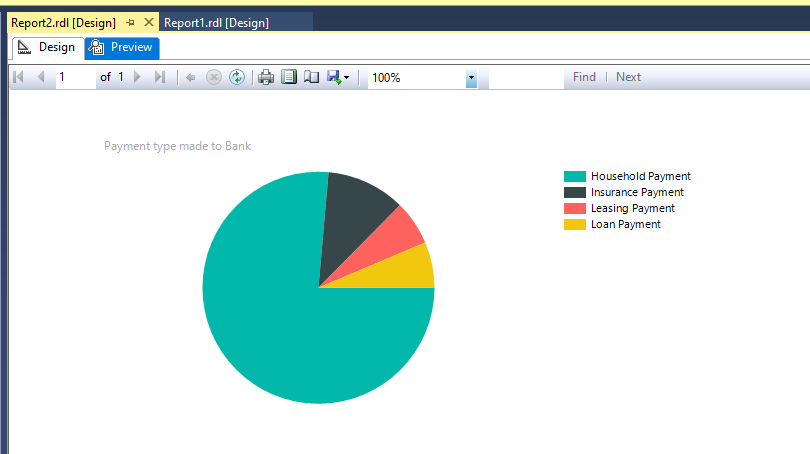




# 4. Reports and Visualization

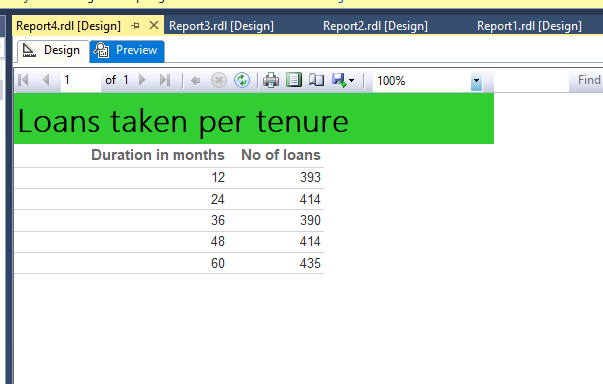
## 4.1 Implementation in SSRS

### 4. 1.1 Transactions made in bank



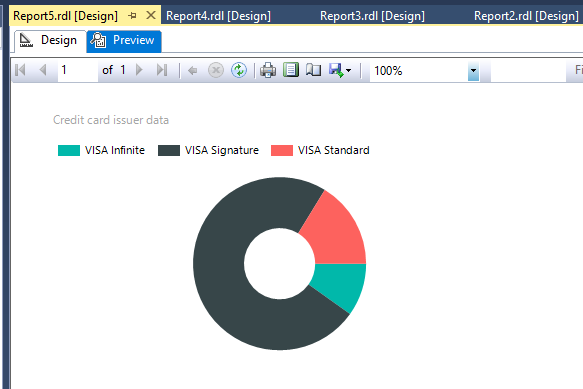
The above chart gives insights about the various payments made by all the customers of the bank.

### 4.1.2 Loans per tenure in months:



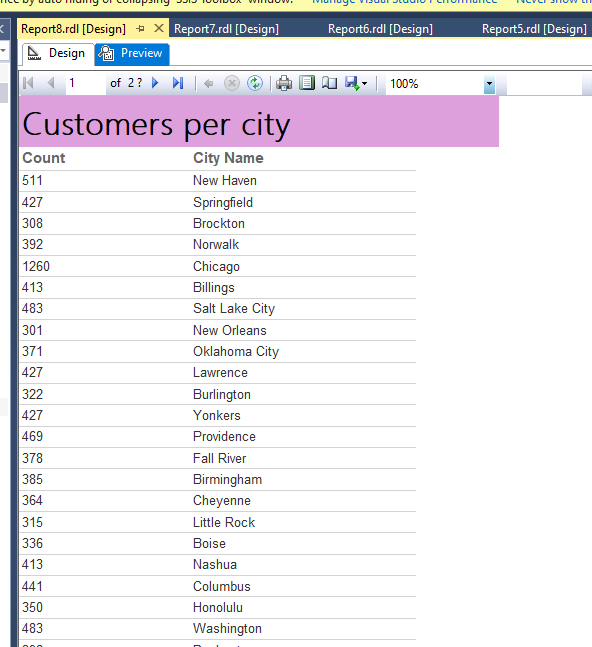
The above table lists the various loan tenures offered to customers and the no of loans provided to each tenure.

### 4.1.3 Credit card type issued to customer



The above graph displays the various cards provided to the customers based on their credit history and profile.

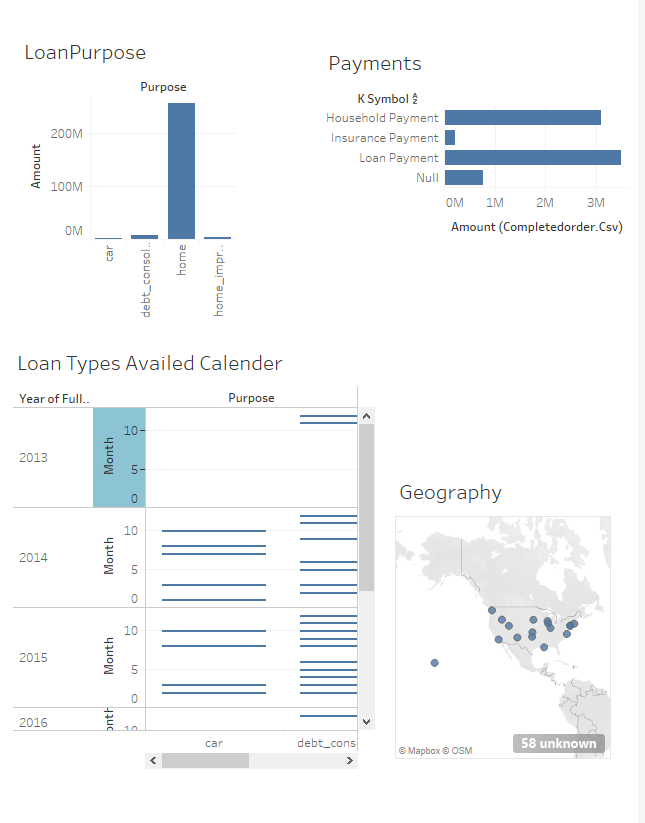
### 4.1.4 Worst affected in low income group:



The above table provides the customers registered on various branches of the bank.

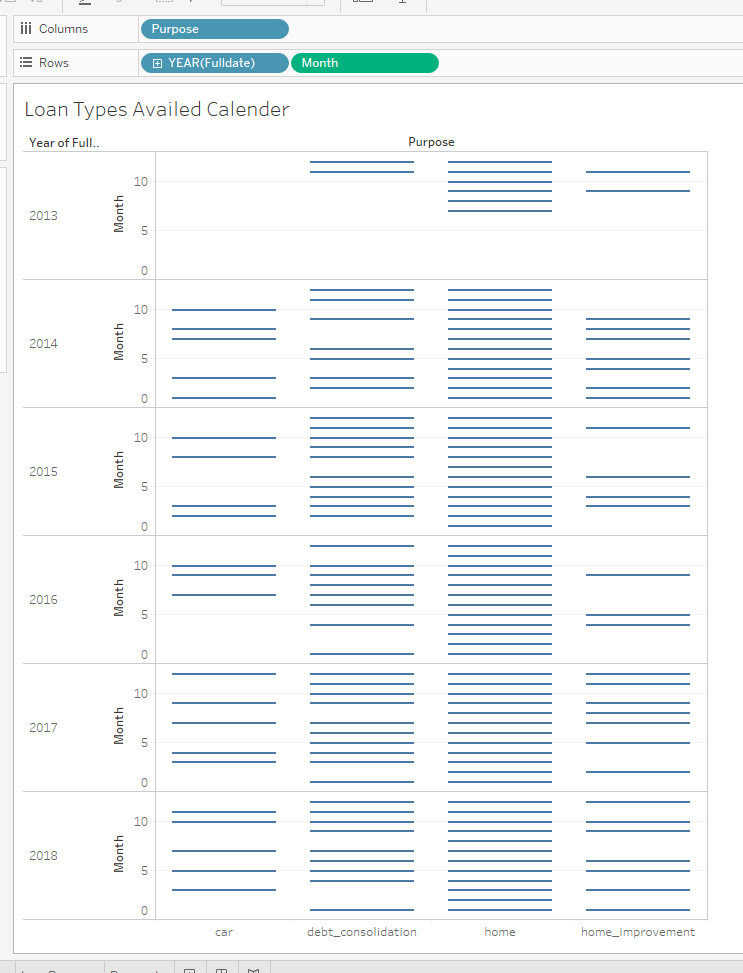
## 4.2 Implementation in Tableau

Dashboard:



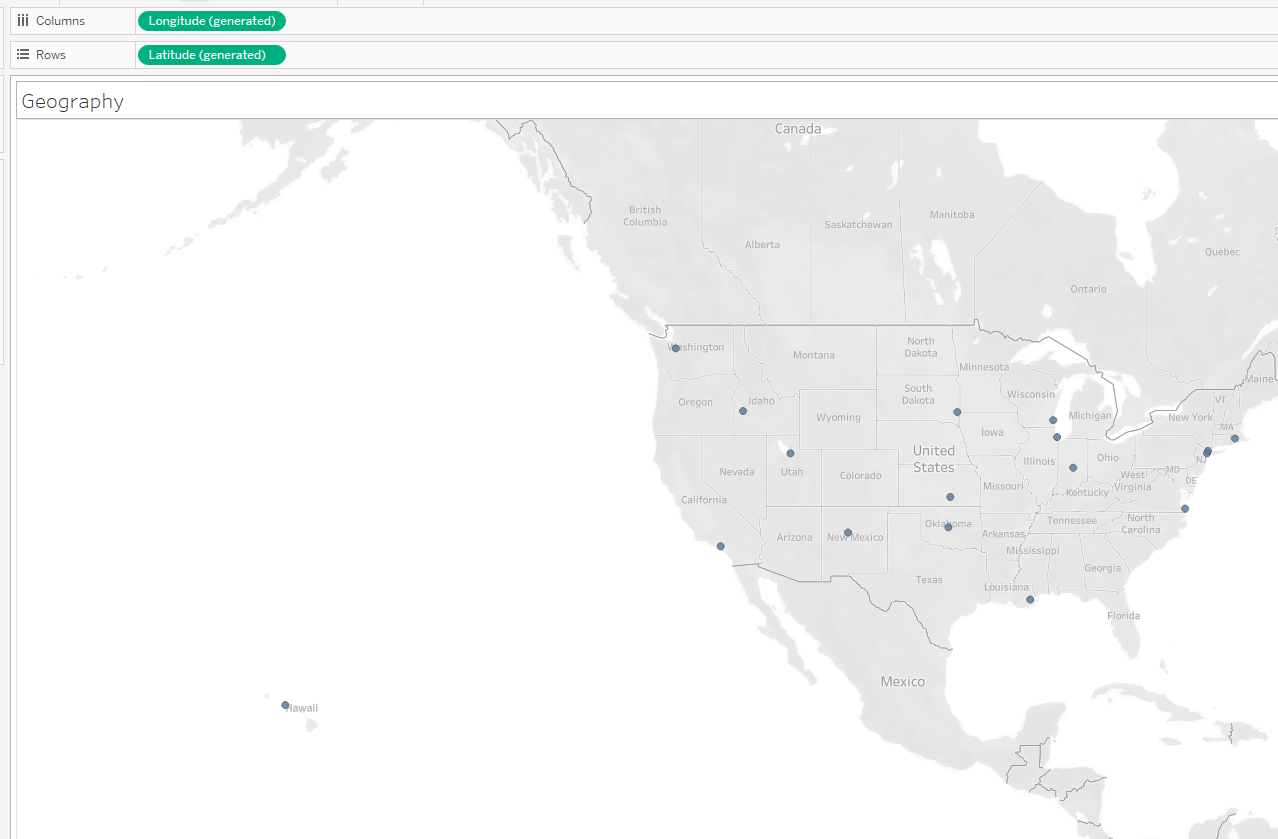
### 4.2.1 Tableau implementation 1: Loan types availed over calender

#### Chart:



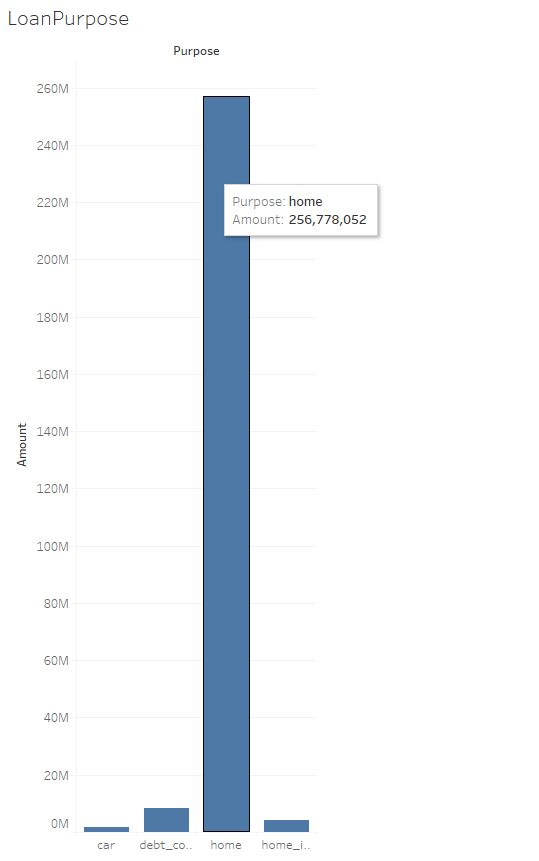
### 4.2.2 Tableau implementation 2: Geography of branches

#### Chart:



### 4.2.3 Tableau implementation 3: loan amount granted based on purpose

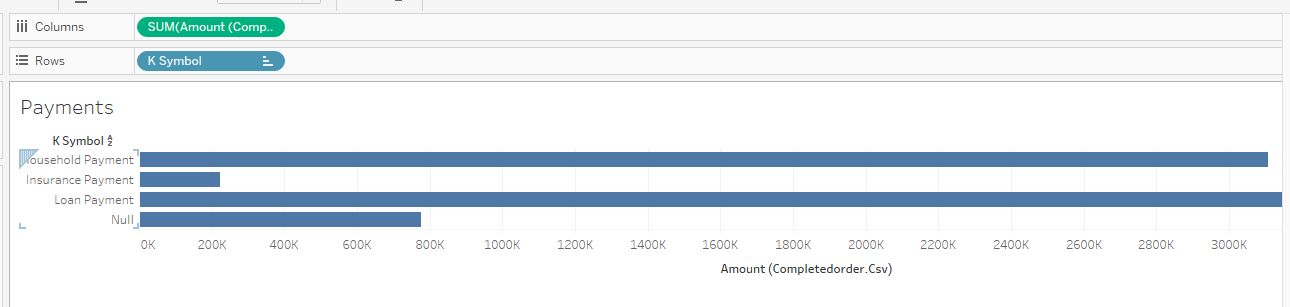
#### Chart:



The gender divide between all the account holders of the bank.

### 4.2.4 Tableau implementation 4: loan payments made

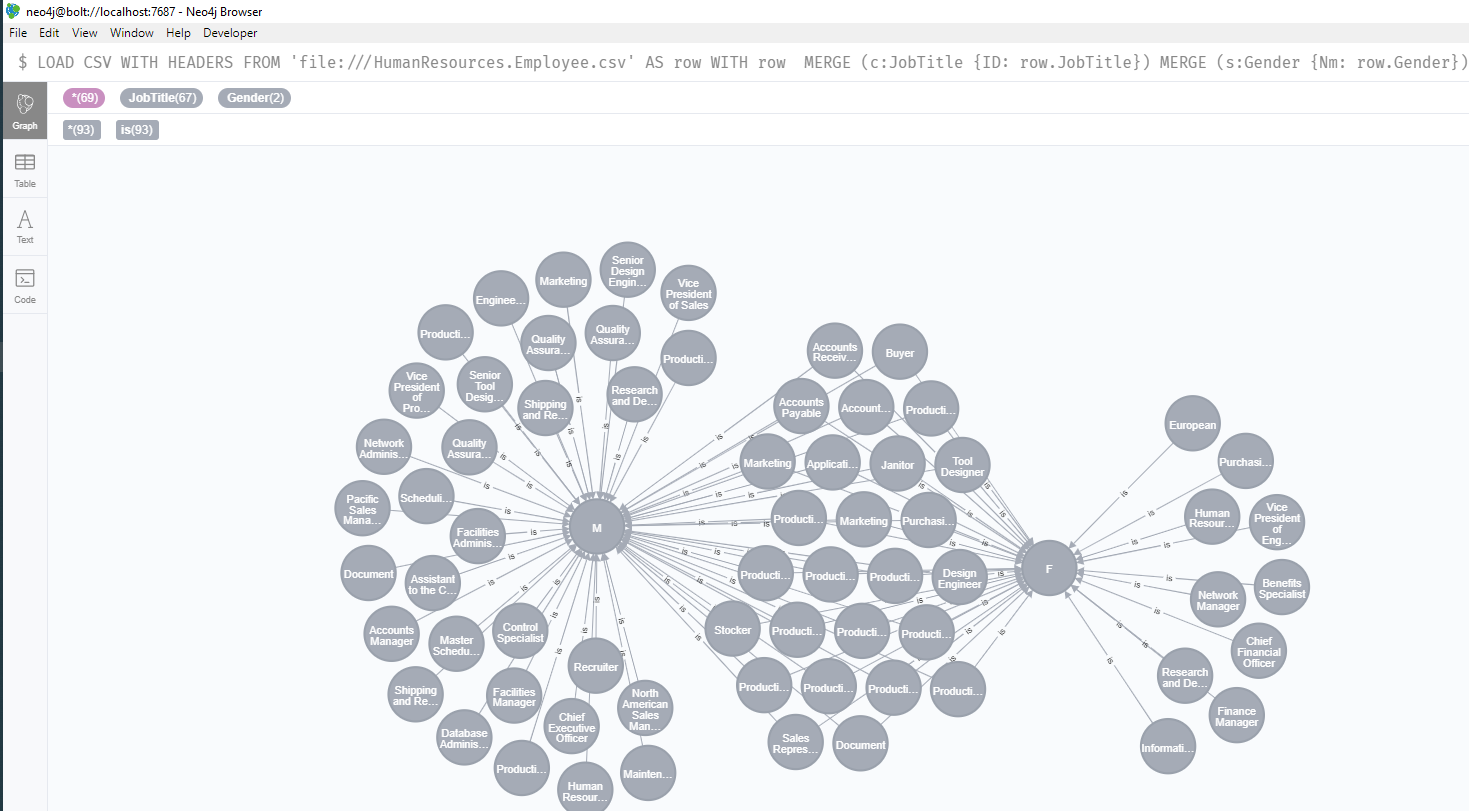
#### Chart:



Loan payments made by customers

# 5. Graph database: neo4j & cypher queries

## 5.1 –relation ship between job title and gender



LOAD CSV WITH HEADERS FROM 'file:///HumanResources.Employee.csv' AS row WITH row

MERGE (c:JobTitle {ID: row.JobTitle})

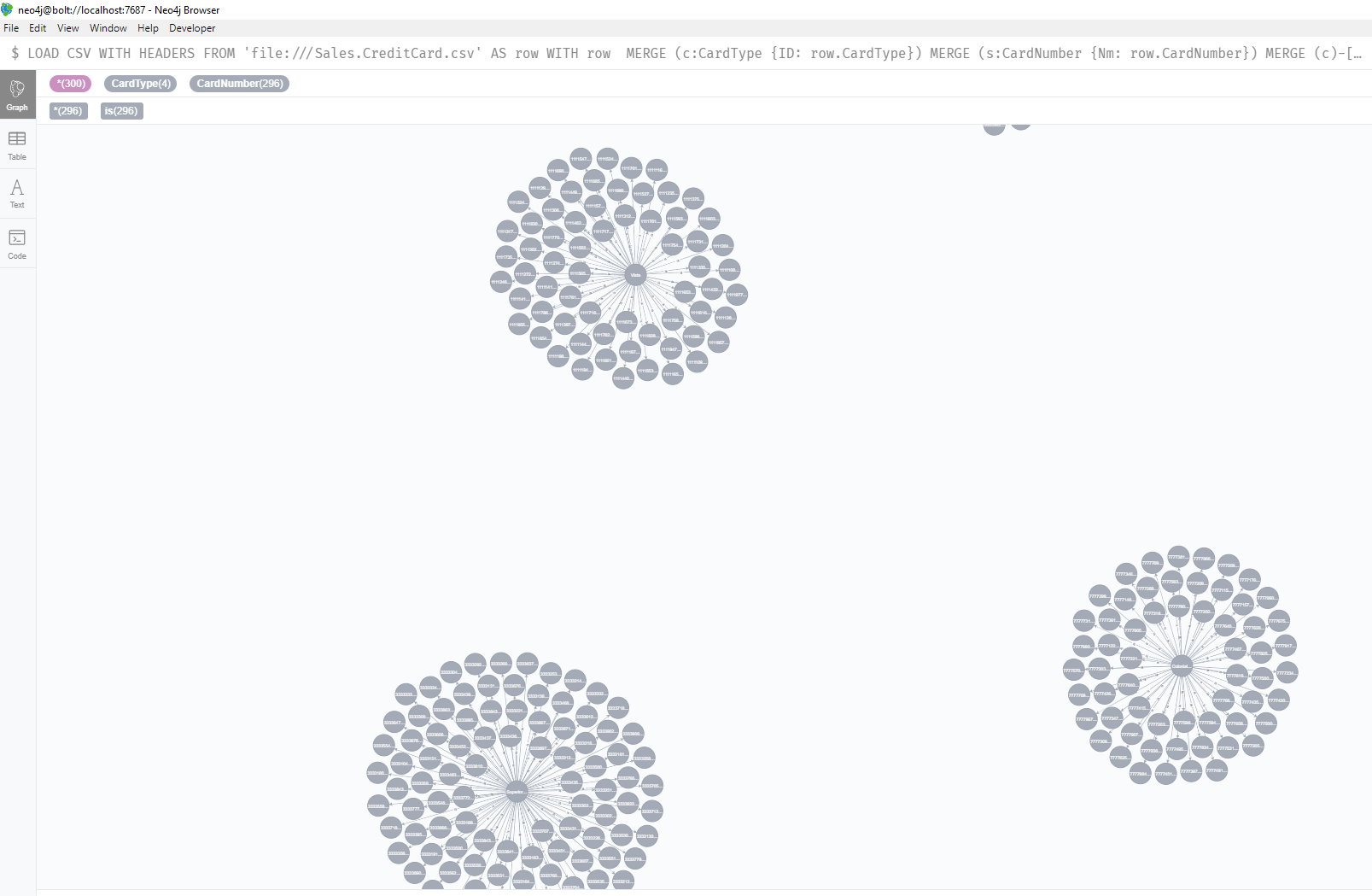
MERGE (s:Gender {Nm: row.Gender})

MERGE (c)-[r:is]->(s)

return c,s

;

## 5.2 –card type subscriptions



LOAD CSV WITH HEADERS FROM 'file:///Sales.CreditCard.csv' AS row WITH row

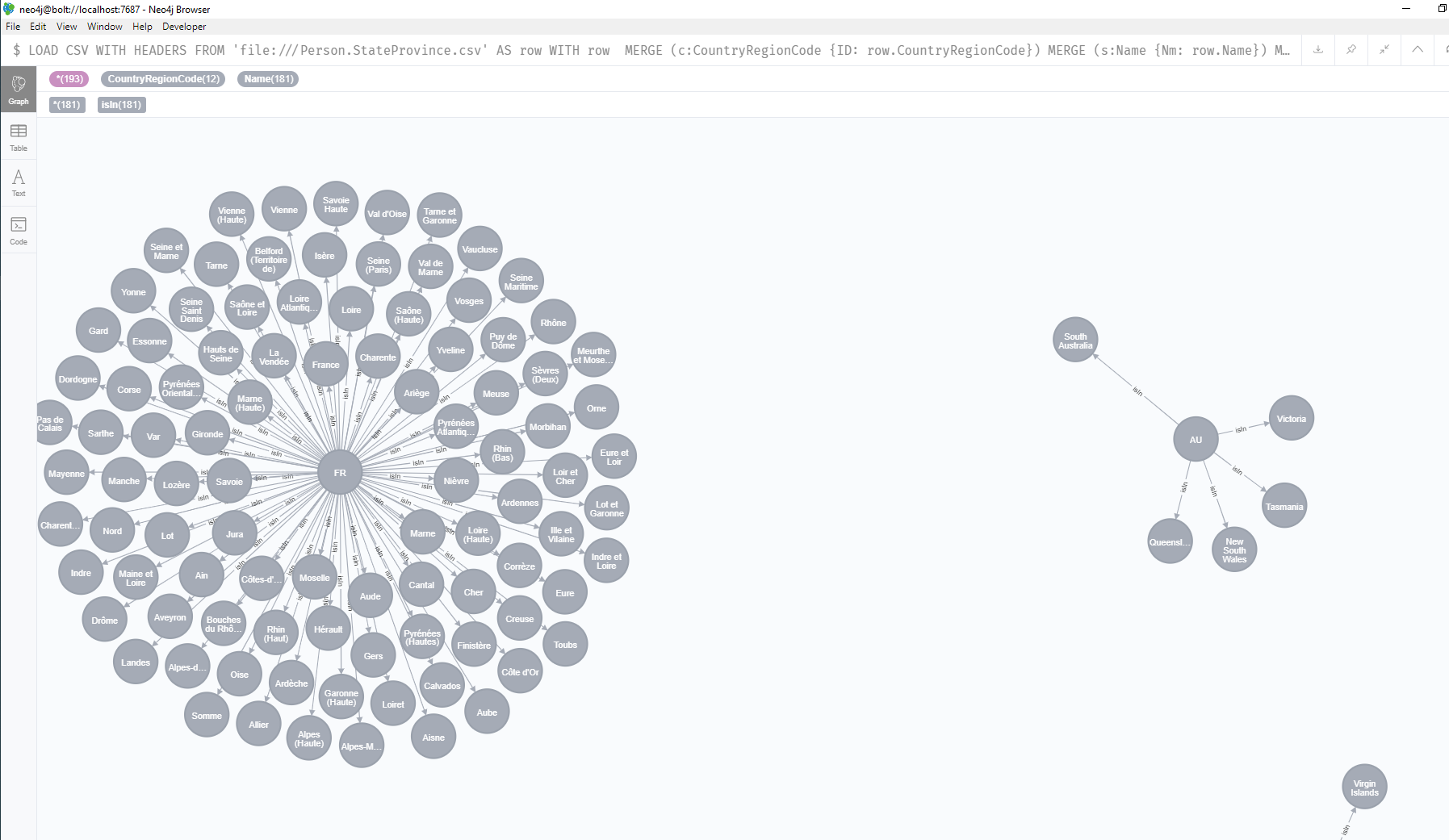
MERGE (c:CardType {ID: row.CardType})

MERGE (s:CardNumber {Nm: row.CardNumber})

MERGE (c)-[r:is]->(s)

return c,s

## 5.3 –delivery location to country relationship



LOAD CSV WITH HEADERS FROM 'file:///Person.StateProvince.csv' AS row WITH row

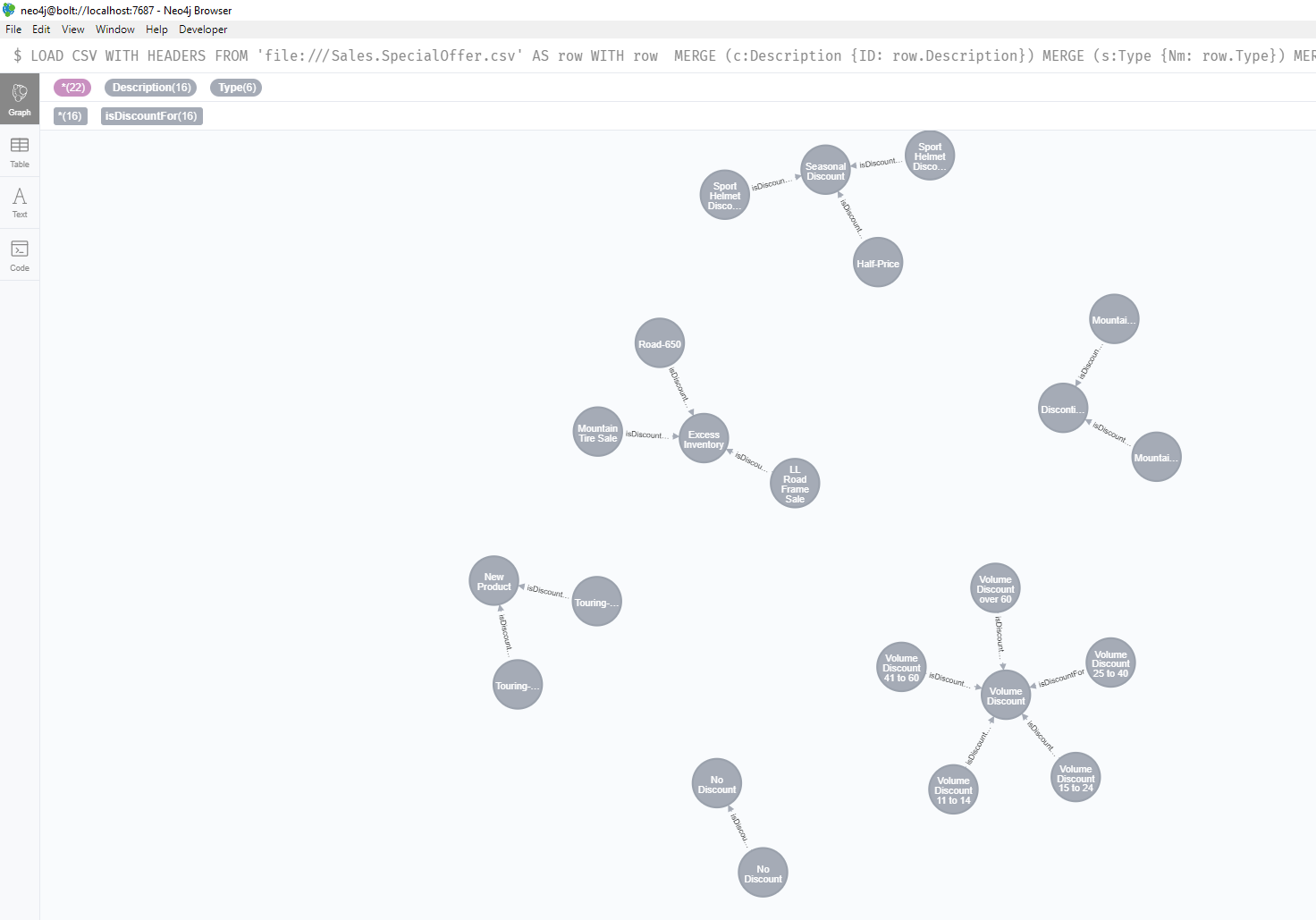
MERGE (c:CountryRegionCode {ID: row.CountryRegionCode})

MERGE (s:Name {Nm: row.Name})

MERGE (c)-[r:isIn]->(s)

return c,s

## 5.4 –Discount Type to category comparison



LOAD CSV WITH HEADERS FROM 'file:///Sales.SpecialOffer.csv' AS row WITH row

MERGE (c:Description {ID: row.Description})

MERGE (s:Type {Nm: row.Type})

MERGE (c)-[r:isDiscountFor]->(s)

return c,s

# 6. Conclusion

Thus, the core banking data set was modelled loaded via SSIS integration tool, visualized through SSRS, R and Neo4J. The data set contained various files related to account, loan, cards, transaction etc. information. Which were data modelled using Toad data modeler and the same is displayed in the above sections. The data is loaded to the Datawarehouse through MSBI provided integration tool SSIS. and SSRS visualization tool provided in MSBI package was used to visualize some graphs. Later the files were parsed through R and graphs were visualized. And finally Neo4J was used to give graphical representation to various relations in the given data.

# 7. Bibliography

Liz petrocelli. 2020. Retail Banking Demo Data . [Online]. [13 April 2020]. Available from: https://data.world/lpetrocelli/retail-banking-demo-data

# APPENDIX A – neo4j CODE

scripts

LOAD CSV WITH HEADERS FROM 'file:///HumanResources.Employee.csv' AS row WITH row

MERGE (c:JobTitle {ID: row.JobTitle})

MERGE (s:Gender {Nm: row.Gender})

MERGE (c)-[r:is]->(s)

return c,s

;

LOAD CSV WITH HEADERS FROM 'file:///Sales.CreditCard.csv' AS row WITH row

MERGE (c:CardType {ID: row.CardType})

MERGE (s:CardNumber {Nm: row.CardNumber})

MERGE (c)-[r:is]->(s)

return c,s

LOAD CSV WITH HEADERS FROM 'file:///Person.StateProvince.csv' AS row WITH row

MERGE (c:CountryRegionCode {ID: row.CountryRegionCode})

MERGE (s:Name {Nm: row.Name})

MERGE (c)-[r:isIn]->(s)

return c,s

LOAD CSV WITH HEADERS FROM 'file:///Sales.SpecialOffer.csv' AS row WITH row

MERGE (c:Description {ID: row.Description})

MERGE (s:Type {Nm: row.Type})

MERGE (c)-[r:isDiscountFor]->(s)

return c,s